fw

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/826,234	04/15/2004	Charles Migos	18602-08908 (P3338US1)	9649

61520 7590 08/22/2007

APPLE/FENWICK
SILLICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW, CA 94041

EXAMINER	
PHAN, TIEN P	
ART UNIT	PAPER NUMBER
2609	
MAIL DATE	DELIVERY MODE
08/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/826,234	MIGOS ET AL.	
	Examiner	Art Unit	
	Tien Phan	2609	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 April 2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 16, 17, 18, 19, and 20 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter; Claim 16, 17, 18, 19, and 20 does not recite that the computer program product comprises a computer readable medium having computer readable program instructions or code embodied thereon and configured to control a computer to perform specific functional steps. The lack of recitation of any specific computer readable medium results in a claim that recites functionally descriptive material (defined as "data structures and computer programs with impart functionality when encoded on a computer readable medium" by the Computer-Implemented Invention Guidelines) without any interrelationships between the data structure and other aspects of the invention that would permit the data structure's functionality to be realized.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claim 1, 2, 10, 14, 15, 16, 19, 20, 21, 24, 25, 26, 29, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), in view of US Patent 5,995,079 by Sheasby et al. ("Sheasby")

As to claim 1, Cooper discloses a method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a control comprising a drag region, an incrementer region, and a decrementer region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); in response to receiving user input placing the graphical pointer within the incrementer region and clicking, incrementing the value of the parameter (See Fig. 1a; where sliders 12-20 that have pointer to the right to increase the value of the parameter); and in response to receiving user input placing the graphical pointer within the decrementer region and clicking, decrementing the value of the parameter (See Fig. 1a; where sliders 12-20 that have pointer to the left to decrease the value of the parameter). Accordingly, Cooper does not expressly teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical

pointer in a second direction, dynamically decrementing the value of the parameter.

However Sheasby teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter (See Fig 1 where it teaches a user may highlight the display numeric value and key a new value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter (See Fig. 1; where user drag the edge 18 to the right will increase the value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter (See Fig. 1; where user drag the edge 18 to the left will decrease the value). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Sheasby, in view of the teaching in the Cooper reference, because the motivation for modifying the references or to combine the reference teachings would allowed the user to manipulate the drag function to change the value of the parameter.

As to claim 2, Cooper discloses a method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box

displaying a value and sliders 12-20 are drag region); and in response to a user command for displaying recent values, displaying a contextual menu of recent values (See Fig. 1a; where 2-10 are text box displaying a recent values). Accordingly, Cooper does not expressly teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter.

However Sheasby teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter (See Fig 1 where it teaches a user may highlight the display numeric value and key a new value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter (See Fig. 1; where user drag the edge 18 to the right will increase the value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter (See Fig. 1; where user

drag the edge 18 to the left will decrease the value). In addition, the same motivation is used as the rejection for claim 1.

As to claim 10, Cooper discloses a method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag region); in response to receiving user input placing the graphical pointer within the drag region to a first side of the text box and a user command for displaying boundaries, displaying a contextual menu to select a upper boundary for the value (See Fig. 1a; where sliders 12-20 that have slider to the right are separate by boundaries); and in response to receiving user input placing the graphical pointer within the drag region to a second side of the text box and user command for displaying boundaries, displaying a contextual menu to select a lower boundary for the value(See Fig. 1a; where sliders 12-20 that have slider to the left are separate by boundaries). Accordingly, Cooper does not expressly teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter; in response to receiving user input placing the graphical pointer within

the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter.

However Sheasby teaches in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter (See Fig 1 where it teaches a user may highlight the display numeric value and key a new value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter (See Fig. 1; where user drag the edge 18 to the right will increase the value); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter (See Fig. 1; where user drag the edge 18 to the left will decrease the value). In addition, the same motivation is used as the rejection for claim 1.

As to claim 14, Cooper discloses a method of receiving user input for changing parameters via a graphical user interface, the method comprising: displaying first and second controls, each control comprising a drag region, each drag region comprising a text box displaying a value for a parameter(See Fig 1b; where there are 2 control are taught by the saturation and hue parameter); in response to receiving user selection of at least two controls, activating the at least two controls(See Fig 1b; where there are 2 control are taught by the saturation and hue parameter); Accordingly, Cooper does not expressly teaches

in response to receiving user input dragging one of the at least two controls, dynamically changing the parameters values corresponding to the at least two controls.

However Sheasby teaches in response to receiving user input dragging one of the at least two controls, dynamically changing the parameters values corresponding to the at least two controls (See Fig 1 where the user can move the pointer to the edge 18 and drag it to the desired position). In addition, the same motivation is used as the rejection for claim 1.

As to claim 15, Cooper discloses in response to receiving user input positioning a graphical pointer within the text box of one of the at least two controls, clicking, and keying in a value, changing the value of the at least two controls (See Fig 1b;where the hue and saturation value can be change by the 54 and 56 controller).

As to claim 16, Cooper discloses a computer program product for receiving user input for changing a parameter via a graphical user interface, comprising: a software portion configured to display a control comprising a drag region, an incrementer region, and a decrementer region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); a software portion configured to increment the value of the parameter in response to receiving user input placing the graphical pointer within the incrementer region and clicking(See Fig. 1a; where sliders 12-20 that have

pointer to the right to increase the value of the parameter); and a software portion configured to decrement the value of the parameter in response to receiving user input placing the graphical pointer within the decrementer region and clicking (See Fig. 1a; where sliders 12-20 that have pointer to the left to decrease the value of the parameter). Accordingly, Cooper does not expressly teaches a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value; a software portion configured to dynamically increment the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction; a software portion configured to dynamically decrement the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction.

However Sheasby teaches a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value (See Fig 1 where it teaches a user may highlight the display numeric value and key a new value); a software portion configured to dynamically increment the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction (See Fig. 1; where user drag the edge 18 to the right will increase the value); a software

portion configured to dynamically decrement the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction (See Fig. 1; where user drag the edge 18 to the left will decrease the value). In addition, the same motivation is used as the rejection for claim 1.

As to claim 19, Cooper discloses a computer program product for receiving user input for changing parameters via a graphical user interface, comprising: a software portion configured to display first and second controls, each control comprising a drag region, each drag region comprising a text box displaying a value for a parameter(See Fig. 1b; where there are 2 control are taught by the saturation and hue parameter); a software portion configured to activate at least two controls in response to receiving user selection of at the least two controls(See Fig. 1b; where there are 2 control are taught by the saturation and hue parameter); Accordingly, Cooper does not expressly teaches a software portion configured to dynamically change the parameter values corresponding to the at least two controls in response to receiving user input dragging one of the at least two controls.

However Sheasby teaches a software portion configured to dynamically change the parameter values corresponding to the at least two controls in response to receiving user input dragging one of the at least two controls (See Fig. 1 where the user may point to the edge 18 and drag the edge to the desire position). In addition, the same motivation is used as the rejection for claim 1.

As to claim 20, Cooper discloses the computer program product comprising: a software portion configured to change the value of the at least two controls in response to receiving user input positioning a graphical pointer within the text box of one of the at least two controls, clicking, and keying in a value (See Fig 1b;where the hue and saturation value can be change by the 54 and 56 controller).

As to claim 21, Cooper discloses a system for receiving user input for changing a parameter via a graphical user interface, comprising:a software portion configured to display a control comprising a drag region, an incrementer region, and a decrementer region, the drag region comprising a text box displaying a value for the parameter(See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); a software portion configured to increment the value of the parameter in response to receiving user input placing the graphical pointer within the incrementer region and clicking (See Fig. 1a; where sliders 12-20 that have pointer to the right to increase the value of the parameter); and a software portion configured to decrement the value of the parameter in response to receiving user input placing the graphical pointer within the decrementer region and clicking (See Fig. 1a; where sliders 12-20 that have pointer to the left to decrease the value of the parameter). Accordingly, Cooper does not expressly teaches a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking,

and keying in a value; a software portion configured to dynamically increment the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction; a software portion configured to dynamically decrement the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction.

However Sheasby teaches a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value (See Fig 1 where it teaches a user may highlight the display numeric value and key a new value); a software portion configured to dynamically increment the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction (See Fig. 1; where user drag the edge 18 to the right will increase the value); a software portion configured to dynamically decrement the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction (See Fig. 1; where user drag the edge 18 to the left will decrease the value). In addition, the same motivation is used as the rejection for claim 1.

As to claim 24, Cooper discloses a system for receiving user input for changing parameters via a graphical user interface, comprising: a software

portion configured to display first and second controls, each control comprising a drag region, each drag region comprising a text box displaying a value for a parameter (See Fig 1b; where there are 2 control are taught by the saturation and hue parameter); a software portion configured to activate at least two controls in response to receiving user selection of at the least two controls (See Fig 1b; where there are 2 control are taught by the saturation and hue parameter). Accordingly, Cooper does not expressly teaches a software portion configured to dynamically change the parameter values corresponding to the at least two controls in response to receiving user input dragging one of the at least two controls.

However Sheasby teaches a software portion configured to dynamically change the parameter values corresponding to the at least two controls in response to receiving user input dragging one of the at least two controls (See Fig. 1 where the user may point to the edge 18 and drag the edge to the desire position). In addition, the same motivation is used as the rejection for claim 1.

As to claim 25, Cooper discloses a software portion configured to change the value of the at least two controls in response to receiving user input positioning a graphical pointer within the text box of one of the at least two controls, clicking, and keying in a value (See Fig 1b;where the hue and saturation value can be change by the 54 and 56 controller).

As to claim 26, Cooper discloses a system for receiving user input for changing a parameter via a graphical user interface, comprising: means for

displaying a control comprising a drag region, an incrementer region, and a decrementer region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); means for incrementing the value of the parameter in response to receiving user input placing the graphical pointer within the incrementer region and clicking (See Fig. 1a; where sliders 12-20 that have pointer to the right to increase the value of the parameter); and means for decrementing the value of the parameter in response to receiving user input placing the graphical pointer within the decrementer region and clicking (See Fig. 1a; where sliders 12-20 that have pointer to the left to decrease the value of the parameter). Accordingly, Cooper does not expressly teaches means for changing the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value; means for dynamically incrementing the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction; means for dynamically decrementing the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction.

However Sheasby teaches means for changing the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value (See Fig 1 where it teaches a user may

highlight the display numeric value and key a new value); means for dynamically incrementing the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction(See Fig. 1; where user drag the edge 18 to the right will increase the value); means for dynamically decrementing the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction (See Fig. 1; where user drag the edge 18 to the left will decrease the value).In addition, the same motivation is used as the rejection for claim 1.

As to claim 29, Cooper discloses a system for receiving user input for changing parameters via a graphical user interface, comprising: means for displaying first and second controls, each control comprising a drag region, each drag region comprising a text box displaying a value for a parameter(See Fig 1b; where there are 2 control are taught by the saturation and hue parameter); means for activating at least two controls in response to receiving user selection of at the least two controls(See Fig 1b; where there are 2 control are taught by the saturation and hue parameter). Accordingly, Cooper does not expressly teaches a means for changing the parameter values corresponding to the at least two controls in response to receiving user input dragging one of the at least two controls .

However Sheasby teaches a means for changing the parameter values corresponding to the at least two controls in response to receiving user input

dragging one of the at least two controls (See Fig 1 where the user can move the pointer to the edge 18 and drag it to the desired position). In addition, the same motivation is used as the rejection for claim 1.

As to claim 30, Cooper discloses a system of claim 29, further comprising: means for changing the value of the at least two controls in response to receiving user input positioning a graphical pointer within the text box of one of the at least two controls, clicking, and keying in a value (See Fig 1b;where the hue and saturation value can be change by the 54 and 56 controller).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 3 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), to US Patent 5,995,079 by Sheasby et al. ("Sheasby"), in further view of US Patent 4,926,360 by Spink,JR. ("Spink").

As to claim 3, Cooper and Sheasby disclose all the limitation of claim 2. Accordingly, Cooper and Sheasby does not expressly teaches the user command for displaying recent values is a click and hold.

However, Spink teaches the user command for displaying recent values is a click and hold(See Fig 11, line 31-40; when you click the measure and hold button will display the measurement of the gage amplifier). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface of Spink, in view of the teaching in the Cooper and Sheasby reference, because the motivation for modifying the references or to combine the reference teachings would enable the user to display the recent parameter values by clicking on the the measure and hold button.

As to claim 11, Spink teaches the user command for displaying recent values is a click and hold(See Fig 11, line 31-40; when you click the measure and hold button will display the measurement of the gage amplifier). In addition, the same motivation is used as the rejection for claim 3.

6. Claim 4 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), to US Patent 5,995,079 by Sheasby et al. ("Sheasby"), in further view of US PGPub US20020059195A1 by Cras et al ("Cras").

As to claim 4, Cooper and Sheasby disclose all the limitation of claim 2. Accordingly, Cooper and Sheasby does not expressly teaches the user command for displaying recent values is a right click .

However, Cras teaches the user command for displaying recent values is a right click(See Fig 10b and c, Paragraph 0070; where right clicking on the

state cell will cause the causes the definition wizard to be display). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Cras, in view of the teaching in the Cooper and Sheasby reference, because the motivation for modifying the references or to combine the reference teachings would enable the user to display the current parameter value by right clicking the graphical pointer on the object.

As to claim 12, Cras teaches the user command for displaying recent values is a right click(See Fig 10b and c, Paragraph 0070; where right clicking on the state cell will cause the causes the definition wizard to be display). In addition, the same motivation is used as the rejection for claim 4.

7. Claim 5 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), to US Patent 5,995,079 by Sheasby et al. ("Sheasby"), in further view of US Patent 5,822,123 by Davis et al ("Davis").

As to claim 5, Cooper and Sheasby disclose all the limitation of claim 2. Accordingly, Cooper and Sheasby does not expressly teaches the user command for displaying recent values is a click combined with a modifier key.

However, Davis teaches the user command for displaying recent values is a click combined with a modifier key(See Column 40, line 38-44, where the user hold down a modifier key can perform different functions). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Davis, in view of

the teaching in the Cooper and Sheasby reference, because the motivation for modifying the references or to combine the reference teachings would enable the user to diplay the recent values of the parameter by holding down the modifier key combine with a click.

As to claim 13, Davis teaches the user command for displaying recent values is a click combined with a modifier key(See Column 40, line 38-44, where the user hold down a modifier key can perform different functions). In addition, the same motivation is used as the rejection for claim 5.

8. Claim 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), in view of US PGPub US20050259077A1 by Adams et al ("Adams").

As to claim 6, Cooper disclose a method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag region); in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter(See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter at a first rate(See Fig. 1a; where sliders 12-20 that

have slider to the right to increase the value of the parameter); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a second direction, dynamically decrementing the value of the parameter at the first rate (See Fig. 1a; where sliders 12-20 that have slider to the left to decrease the value of the parameter); Accordingly, Cooper does not expressly teaches in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer a third direction, dynamically incrementing the numeric value at a second rate; in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a fourth direction, dynamically decrementing the numeric value at the second rate.

However, Adams teaches in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer a third direction, dynamically incrementing the numeric value at a second rate (See paragraph 0032; where by pressing the Alt and F key could increase the rate at which the image is display); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer in a fourth direction, dynamically decrementing the numeric value at the second rate (See paragraph 0032; where by pressing the Alt and F key could increase the rate at which the image is display). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Adams, in view of the teaching in the

Cooper reference, because the motivation for modifying the references or to combine the reference teachings would enable to the user to select a key combining with a modifier key to allow different rate of increment or decrement.

9. Claim 7,17 ,22 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), in view of US PGPub US20050259077A1 by Czerwinski et al ("Czerwinski").

As to claim 7, Cooper disclose a method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a first control comprising a drag region, the drag region comprising a text box displaying a value for the parameter(See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value, changing the value of the parameter (See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a first axis, dynamically changing the value of the parameter(See Fig. 1a; where sliders 12-20 that have slider to the change the value of the parameter); and in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis(See Fig. 1a; where sliders 12-20 that have slider change the value of the parameter).

Accordingly, Cooper does not expressly teaches recognizing the input as an attempt by the user to drag and drop the value into second control.

However, Czerwinski teaches recognizing the input as an attempt by the user to drag and drop the value into second control (See Paragraph 0038, where by utilizing the drag and drop functionality allowed the manipulation of control tiles and control tiles group). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Czerwinski, in view of the teaching in the Cooper reference, because the motivation for modifying the references or to combine the reference teachings would allowed the user to manipulate the control tiles by using the drag and drop functionality.

As to claim 17, Cooper disclose a computer program product for receiving user input for changing a parameter via a graphical user interface, comprising: a software portion configured to display a first control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value (See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); a software portion configured to dynamically change the value of the parameter in response to receiving user input placing the graphical pointer within the drag region,

clicking, and dragging the graphical pointer along a first axis (See Fig. 1a; where sliders 12-20 that have slider to the change the value of the parameter); Accordingly, Cooper does not expressly teaches a software portion configured to recognize an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis.

However, Czerwinski teaches a software portion configured to recognize an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis. (See Paragraph 0038, where by utilizing the drag and drop functionality allowed the manipulation of control tiles and control tiles group). In addition, the same motivation is used as the rejection for claim 7.

As to claim 22, Cooper discloses a system for receiving user input for changing a parameter via a graphical user interface, comprising: a software portion configured to display a first control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); a software portion configured to change the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying in a value (See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); a software

portion configured to dynamically change the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a first axis (See Fig. 1a; where sliders 12-20 that have slider to the change the value of the parameter); Accordingly, Cooper does not expressly teaches a software portion configured to recognize an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis.

However, Czerwinski teaches a software portion configured to recognize an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis. (See Paragraph 0038, where by utilizing the drag and drop functionality allowed the manipulation of control tiles and control tiles group). In addition, the same motivation is used as the rejection for claim 7.

As to claim 27, Cooper discloses a system for receiving user input for changing a parameter via a graphical user interface, comprising: means for displaying a first control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box displaying a value and sliders 12-20 are drag, increment, and decrement region); means for changing the value of the parameter in response to receiving user input positioning a graphical pointer within the text box, clicking, and keying

in a value (See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); means for dynamically changing the value of the parameter in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a first axis (See Fig. 1a; where sliders 12-20 that have slider to the change the value of the parameter); Accordingly, Cooper does not expressly teaches a means for recognizing an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis.

However, Czerwinski teaches a means for recognizing an attempt by the user to drag and drop the value into second control in response to receiving user input placing the graphical pointer within the drag region, clicking, and dragging the graphical pointer along a second axis (See Paragraph 0038, where by utilizing the drag and drop functionality allowed the manipulation of control tiles and control tiles group). In addition, the same motivation is used as the rejection for claim 7.

10. Claim 8,18, 23 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), to US PGPub US20050259077A1 by Czerwinski et al ("Czerwinski"), in further view of US Patent 5,920,477 by Hoffberg et al. ("Hoffberg")

As to claim 8, Cooper and Czerwinski disclose all the limitation of claim 7. Accordingly, Cooper and Czerwinski does not expressly teaches the recognizing

the input occurs only when the dragging the graphical pointer along the second axis exceeds a tolerance angle from the first axis.

However, Hoffberg teaches the recognizing the input occurs only when the dragging the graphical pointer along the second axis exceeds a tolerance angle from the first axis (See column 39, line 9-14, where and column 39, line 1-6, where an input device which provide input information up to six axis which track an object where many degree of freedom are required). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Hoffberg, in view of the teaching in the Cooper and Czerwinski reference, because the motivation for modifying the references or to combine the reference teachings would enable the user to have a degree of accuracy when dragging the pointer to exceed the tolerance angle of the first and second axis.

As to claim 18, Hoffberg teaches the computer program product wherein the software portion configured to recognize an attempt by the user to drag and drop the value into second control recognizes only drags of the graphical pointer along the second axis greater than a tolerance angle from the first axis (See column 39, line 9-14, where and column 39, line 1-6, where an input device which provide input information up to six axis which track an object where many degree of freedom are required). In addition, the same motivation is used as the rejection for claim 8.

As to claim 23, Hoffberg teaches the system wherein the software portion configured to recognize an attempt by the user to drag and drop the value into second control recognizes only drags of the graphical pointer along the second axis greater than a tolerance angle from the first axis (See column 39, line 9-14, where and column 39, line 1-6, where an input device which provide input information up to six axis which track an object where many degree of freedom are required). In addition, the same motivation is used as the rejection for claim 8.

As to claim 28, Hoffberg teaches the system wherein the means for recognizing an attempt by the user to drag and drop the value into second control recognizes only drags of the graphical pointer along the second axis greater than a tolerance angle from the first axis (See column 39, line 9-14, where and column 39, line 1-6, where an input device which provide input information up to six axis which track an object where many degree of freedom are required). In addition, the same motivation is used as the rejection for claim 8.

11. Claim 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,477,271b1 by Cooper et al ("Cooper"), in view of US Patent 6,340,966 by Wang et al ("Wang").

As to claim 9, Cooper disclose A method of receiving user input for changing a parameter via a graphical user interface, the method comprising: displaying a control comprising a drag region, the drag region comprising a text box displaying a value for the parameter (See Fig. 1a; where 2-10 are text box

displaying a value and sliders 12-20 are drag, increment, and decrement region); in response to receiving user input hovering a graphical pointer over the text box and keying in a value, changing the value of the parameter(See Column 5, line 34-44; where it teaches a plurality of controls for changing parameters); in response to receiving user input hovering the graphical pointer over the drag region and dragging the graphical pointer in a first direction, dynamically incrementing the value of the parameter (See Fig. 1a; where sliders 12-20 that have slider to the right to increase the value of the parameter); Accordingly, Cooper does not expressly teaches in response to receiving user input hovering the graphical pointer over the drag region and scrolling a mouse wheel in a first direction, dynamically incrementing the value of the parameter; and in response to receiving user input hovering the graphical pointer over the drag region and scrolling a mouse wheel in a second direction, dynamically decrementing the value of the parameter.

However, Wang teaches in response to receiving user input hovering the graphical pointer over the drag region and scrolling a mouse wheel in a first direction, dynamically incrementing the value of the parameter (See Column 3, line 1-5, where a mouse is added with a third axis input device for scrolling function); and in response to receiving user input hovering the graphical pointer over the drag region and scrolling a mouse wheel in a second direction, dynamically decrementing the value of the parameter (See Column 3, line 1-5, where a mouse is added with a third axis input device for scrolling function).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the graphical user interface in the system of Wang, in view of the teaching in the Cooper reference, because the motivation for modifying the references or to combine the reference teachings would allowed the user to used the scrolling function of the mouse to hover the graphical pointer over the drag region to either increment or decrement the vlue of the parameter.

Conclusion

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tien Phan whose telephone number is 571 270 3309. The examiner can normally be reached on Monday-Friday 7:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Derrick Ferris can be reached on 571 272 3123. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

8/16/07

DERRICK W. FERRIS
SUPERVISORY PATENT EXAMINER